

Application No.: 09/846,645

Docket No.: 10004560-1

AMENDMENTS TO THE CLAIMS

1. (Original) A method for determining a type of device associated with an input/output (I/O) path, comprising:
 - retrieving device information from a device associated with said I/O path utilizing a device control protocol;
 - retrieving a property file defining a type of device;
 - executing code associated with said property file, wherein said code is operable to determine whether said device is said type of device utilizing in part said retrieved device information.
2. (Original) The method of claim 1 wherein said property file identifies a class defining said type of device, said method further comprising:
 - instantiating an object of said class;
 - wherein said step of executing code includes calling a method of said instantiated object.
3. (Original) The method of claim 2 wherein said executing code determines that said device is said type of device, said method further comprising:
 - calling a second method of said instantiated object to create a unique identifier for said device.
4. (Previously Presented) The method of claim 1 wherein said device is a small computer system interface (SCSI) device, and wherein said step of retrieving said device information includes obtaining a vendor identifier and a product identifier of said device from a host agent.
5. (Original) The method of claim 1 wherein said device is an simple network management protocol (SNMP) device, and wherein said step of retrieving includes obtaining a SNMP system object identifier of said device.
6. (Original) The method of claim 1 wherein said code is operable to query said device to determine whether said device is said type of device.

Application No.: 09/846,645

Docket No.: 10004560-1

7. (Original) A system for discovery a type of device associated with an input/output (I/O) path, comprising:
at least one processor for executing code;
means for defining a type of device;
means for obtaining device information associated with said I/O path; and
code operable to determine whether said device associated with said I/O path is said type of device utilizing in part said device information obtained by said means for obtaining and information obtained from said means for defining.

8. (Original) The system of claim 7 wherein said system further comprises:
means for assigning a unique identifier to said device.

9. (Original) A system for discovering a device associated with an input/output (I/O) path, comprising:
at least one processor for executing code;
a plurality of data structures wherein each of said data structure respectively defines a type of device;
code for removing a class identifier from each of said data structures, wherein said class identifier identifies a respective class;
code for instantiating an object of said respective class of each class identifier; and
code for calling a method of each instantiated object, wherein said method is operable to determine whether a device associated with said I/O path is the type of device defined by said data structure associated with said respective instantiated object of said method.

10. (Original) The system of claim 9 wherein said code for instantiating an object further comprises:

code for creating an array of handles to said instantiated objects.

11. (Original) The system of claim 10 wherein said code for calling comprising:
code for removing a handle from said array of handles.

12. (Original) The system of claim 10 wherein said method of each object instance is operable to communicate with a host agent to obtain information utilized to determine whether said device is the type of device associated with the respective data structure.

Application No.: 09/846,645

Docket No.: 10004560-I

13. (Original) The system of claim 12 wherein said host agent provides an application programming interface (API) to obtain said information.

14. (Previously Presented) A method for discovering a type of device associated with an input/output (I/O) path of a storage area network, comprising:

- (a) retrieving a plurality of property files from a predefined subdirectory, wherein each property file of said plurality of property files describes a type of device;
- (b) removing a class identifier from each property file of said property files, wherein each class identifier identifies a class;
- (c) creating an object of the respective class of each class identifier; and
- (d) calling a method of each created object, wherein said method is operable to determine whether a device associated with said I/O path is the type of device described by the property file associated with said object method.

15. (Original) The method of claim 14 further comprising:

- (e) adding a new storage device to said storage area network, wherein said new storage device is caused to be associated with said I/O path, and wherein said new storage device is a new type of device to said storage area network;
- (f) storing a new property file in said predefined subdirectory describing said new type of device; and
- (g) restarting code of a management server to thereby cause repetition of steps (a)-(d) utilizing said new property file.

16. (Original) The method of claim 14 wherein a default property file of said plurality of property files identifies a default small computer system interface (SCSI) class, wherein said default SCSI class defines a method to identify devices by comparing SCSI vendor identifier and product identifier information to at least one field in said default property file.

17. (Original) The method of claim 14 wherein a default property file of said plurality of property files identifies a simple network management protocol (SNMP) class, wherein said default SNMP class defines a method to identify devices by comparing a SNMP system object identifier to at least one field in said default property file.

Application No.: 09/846,645

Docket No.: 10004560-1

18. (Original) A system for analyzing input/output (I/O) paths of a storage area network (SAN) comprising:

a plurality of servers, wherein said servers are communicatively coupled to a fabric of said SAN;

a plurality of host agent processes, wherein each of said host agent processes executes on a respective server of said plurality of servers, and wherein said host agent processes are operable to query devices associated with host logical unit numbers I/O paths of said SAN to gather device information;

a management server, wherein said management server employs a simple network management protocol (SNMP) manager process to query devices associated with SNMP I/O paths of said SAN to gather device information;

a plurality of property files stored in a predefined directory, wherein each property file of said plurality of property files describes a type of device, and wherein each property file of said plurality of property files includes an identifier of code operable to determine whether a device associated with an I/O path is the type of device described by its associated property file; and

a management server process, wherein said management server process is operable to receive gathered device information from said plurality of host agent processes and from said SNMP manager process; and wherein said management server process is operable to call code identified by property files with gathered device information as arguments to thereby identify types of devices associated with I/O paths of said SAN.

19. (Original) The system of claim 18 wherein said management server process, includes:

code for creating an array of identifiers including each said identifier from each property file;

code for instantiating a plurality of small computer system interface (SCSI) device discovery objects utilizing identifiers from said array that identify SCSI device classes; and

code for instantiating a plurality of SNMP device discovery objects utilizing identifiers from said array that identify SNMP device classes.

Application No.: 09/846,645

Docket No.: 10004560-1

20. (Original) The system of claim 19 wherein said management server process includes:

code for calling a method of each instantiated SCSI device discovery object for each host logical unit numbers I/O path; and

code for calling a method of each instantiated SNMP device discovery object for each SNMP I/O path.